**Assessment Time Table**

****

**23/10/2024**

**OOP’S Assignment**

1. What is java and explain feature of java?

2. Why java is platform ind ependent language?

3. What is byte code and importance of byte code?

4. What is difference between byte code and machine code?

5. What is diff between platform independency and cross platform?

6. What is OOP and explain depth?

7. What are the pillars of OOP?

8. What is diff between semi object oriented and pure object oriented?

9. What is JDK, JRE and JVM?

10. What is diff between JDK JRE and JVM?

11. What is JVM and explain its architecture?

12. What is array and how many ways to declare array in java?

13. What is diff between C array and java array?

14. What is Jagged Array in java?

15. What is anonymous array in java? Q1. what is inheritance and why use it?

16. explain types of inheritance in java?

17 Can we create java program without inheritance?

18. Why Object class is parent of every class in java?

19. is true if parent contain default constructor, then it is executed before child?

20 is true if parent contain parameterized constructor, then programmer must be pass parameter

from child?

21. what is super() constructor and why use it?

22. is it possible super() constructor write on second line of code?

23. is it possible super() and this() use at same time?

24. is it true super() pass parameter to immediate parent class?25. Can we implement interface in class?

26. how to solve diamond problem using interface?

27. how to achieve multiple inheritance using interface?

28. what is diff between interface and abstract class?

29. Can we override static method?

30. what is method hiding in java and explain in detail?

31. what is diff between hiding and overriding?

32. what is dynamic polymorphism and how we can achieve in java with code?

33. What is coupling in java and explain detail?

34. What is IS-A relationship and what is HAS-A relationship in java?

PROGRAMS

**Q 1 ) Prime Number Checker for 1 to N**

**Specification**: Create a class **PrimeChecker** that takes an integer N via the constructor and checks if each number from 1 to N is a prime number using a loop.

**Input**: An integer N.

**Output**: Whether each number from 1 to N is a prime number or not.

===========================================================================

**Q 2 Write a java program to create class name as Area using Constructor overloading name as:**

Area(int) {

// calculate area of circle.

} Area(int , int) { // calculate area of reactangle. }

Area(int , float) { // calculate area of triangle}

==================================================================================

**Q 3 ) Assignment: String Compression**

**Objective**

The goal of this assignment is to develop a method for compressing a string by replacing consecutive identical characters with the character followed by the number of occurrences. This exercise will enhance your understanding of string manipulation and algorithm design.

**Problem Statement**

You are required to implement a string compression algorithm. The algorithm should compress a given string by replacing sequences of the same character with a single instance of the character followed by the number of times it appears consecutively.

For example:

* The string "aabcccccaaa" should be compressed to "a2b1c5a3".
* If the compressed string does not reduce the size of the original string, you should return the original string instead.

**Requirements**

1. **Input Handling:**
   * The method should accept a single string as input. If the input string is null or empty, the method should return the string as is.
2. **Compression Logic:**
   * Traverse the input string while counting consecutive occurrences of each character.
   * Append each character and its count to a new string.
   * Continue this process until you reach the end of the string.
3. **Output:**
   * Return the compressed string if it is shorter than the original. If the compressed string is not shorter than the original, return the original string.

***Examples***

1. **Example 1:**
   * Input: "aabcccccaaa"
   * Output: "a2b1c5a3"
2. **Example 2:**
   * Input: "abcd"
   * Output: "a1b1c1d1" (Each character is unique)
3. **Example 3:**
   * Input: "" (Empty String)
   * Output: ""

==================================================================================

**Q 4 : Assignment: Custom Class Sorting**

***Objective***

The purpose of this assignment is to develop an understanding of how to create custom classes and implement sorting based on specific attributes of those classes. You will enhance your skills in object-oriented programming and sorting algorithms.

**Problem Statement**

You are required to implement a method that sorts an array of objects based on a specific attribute of the class.

**Task:**

1. **Define a Class:**
   * Create a class named Student that has the following attributes:
     + name (String): The name of the student.
     + age (int): The age of the student.
     + grade (double): The grade of the student.
   * Include a constructor in the Student class that initializes the attributes.
2. **Create an Array of Objects:**
   * Create an array of Student objects and populate it with at least five instances of the Student class, each with different values for name, age, and grade.
3. **Implement a Sorting Method:**
   * Write a method named sortStudents that takes an array of Student objects and a string attribute (e.g., "name", "age", or "grade") as parameters.
   * The method should sort the array of Student objects based on the specified attribute in ascending order. If the attribute is not recognized, return the array as is.
4. **Display the Results:**
   * After sorting the array, print out the details of each Student object in the sorted

==================================================================================

**Q 5: Assignment: Move All Zeros to the End of an Array**

**Objective**

The goal of this assignment is to implement a method that moves all zeros in a given array to the end while maintaining the order of non-zero elements. You will utilize object-oriented programming concepts by creating a custom class and its constructor.

**Problem Statement**

You are required to create a Java program that moves all zeros in a given array to the end without changing the order of the other elements. To achieve this, you will need to implement a custom class to represent the array and a method to perform the operation.

**Task:**

1. **Define a Class:**
   * Create a class named ArrayManipulator that has the following attributes:
     + numbers (int[]): An array of integers.
   * Include a constructor that initializes the numbers attribute.
2. **Implement a Method:**
   * Write a method named moveZerosToEnd within the ArrayManipulator class that performs the following:
     + Iterate through the numbers array and move all non-zero elements to the front of the array, while counting the number of zeros.
     + After processing the array, fill the remaining positions with zeros.
     + The method should modify the numbers array in place.
3. **Display the Results:**
   * Create a method named displayArray that prints the elements of the numbers array in a readable format.

==================================================================================

**1 Statement: Employee Management System**

**Objective:** Develop a system to manage employees and calculate their salaries.

**Abstract Class Definition:**

* + Create an abstract class named Employee with the following abstract methods:
    - calculateSalary(): This method should return the salary of the employee as a double.
    - displayInfo(): This method should print the details of the employee.

**Subclasses Implementation:**

* + Implement two subclasses of Employee:
    - **FullTimeEmployee**
      * Add a constructor that takes the employee's name and monthly salary as parameters.
      * Implement the calculateSalary() method to return the monthly salary.
      * Implement the displayInfo() method to display the employee's name and monthly salary.
    - **PartTimeEmployee**
      * Add a constructor that takes the employee's name and hourly wage along with the number of hours worked per week as parameters.
      * Implement the calculateSalary() method to return the weekly salary (hourly wage multiplied by hours worked).
      * Implement the displayInfo() method to display the employee's name, hourly wage, and weekly salary.

**Array of Employees:**

* + Create an array of Employee objects that includes at least one FullTimeEmployee and one PartTimeEmployee.
  + Loop through the array to:
    - Call the displayInfo() method on each employee.
    - Call the calculateSalary() method and display the salary.

**Example Output:**

Full-Time Employee: John Doe

Monthly Salary: $3000.00

Part-Time Employee: Jane Smith

Hourly Wage: $20.00

Weekly Salary: $200.00

**Requirements:**

* Ensure proper use of inheritance and abstraction in your implementation.
* Use appropriate access modifiers (public, protected, etc.) where necessary.
* The program should be designed to easily add more employee types in the future.

**2) Appliance Power Management**

**Objective:** Create a system to manage different types of appliances and their power consumption.

**Abstract Class Definition:**

* + Create an abstract class named Appliance with the following:
    - A method turnOn() that simulates turning on the appliance.
    - An abstract method getPowerConsumption() that returns the power consumption of the appliance in watts.

**Subclasses Implementation:**

* + Implement two subclasses of Appliance:
    - **WashingMachine**
      * Implement the turnOn() method to print a message indicating the washing machine is running.
      * Implement the getPowerConsumption() method to return a fixed power consumption value (e.g., 500 watts).
    - **Refrigerator**
      * Implement the turnOn() method to print a message indicating the refrigerator is cooling.
      * Implement the getPowerConsumption() method to return a fixed power consumption value (e.g., 150 watts).

**Array of Appliances:**

* + Create an array of Appliance objects that includes at least one WashingMachine and one Refrigerator.
  + Loop through the array to:
    - Call the turnOn() method on each appliance.
    - Call the getPowerConsumption() method and display the power consumption of each appliance.

**Example Output:**

Washing Machine is now running.

Power Consumption: 500 watts

Refrigerator is now cooling.

Power Consumption: 150 watts

**Requirements:**

* Ensure proper use of inheritance and abstraction in your implementation.
* Use appropriate access modifiers (public, protected, etc.) where necessary.
* The program should be easy to extend for additional appliance types in the future.

**3)Sorting Arrays with Sortable Interface**

Objective: Implement an interface and two classes to sort arrays of integers and strings.

Requirements:

Define an Interface:

Create an interface named Sortable with a method sort().

Implementing Classes:

Create a class named IntegerArray that:

Contains a private array of integers.

Implements the Sortable interface and provides the implementation for the sort() method to sort the integer array in ascending order.

Includes a method printArray() to display the contents of the integer array.

Create a class named StringArray that:

Contains a private array of strings.

Implements the Sortable interface and provides the implementation for the sort() method to sort the string array in alphabetical order.

Includes a method printArray() to display the contents of the string array.

Demonstration:

In the main method of a separate class, create an instance of IntegerArray with an array of integers.

Create an instance of StringArray with an array of strings.

Call the sort() method on both instances to sort the arrays.

Print the sorted arrays using the printArray() method.

Expected Output:

The program should display the sorted integer and string arrays when executed. For example:

Sorted Integer Array: [1, 2, 3, 5, 8]

Sorted String Array: [apple, banana, kiwi, orange]

**4) Notification System with Notifiable Interface**

Objective: Implement an interface and two classes to demonstrate a notification system that can send different types of notifications.

Requirements:

Define an Interface:

Create an interface named Notifiable with a method sendNotification().

Implementing Classes:

Create a class named EmailNotification that:

Implements the Notifiable interface.

Contains a private field to store the notification message.

Implements the setMessage(String message) method to set the content of the email notification.

Implements the sendNotification() method to display the message in the format: "Sending Email: [message]".

Create a class named SMSNotification that:

Implements the Notifiable interface.

Contains a private field to store the notification message.

Implements the setMessage(String message) method to set the content of the SMS notification.

Implements the sendNotification() method to display the message in the format: "Sending SMS: [message]".

Demonstration:

In the main method of a separate class, create an array of Notifiable objects.

Create instances of EmailNotification and SMSNotification, and add them to the array.

Use the setMessage(String message) method to set different messages for each notification type.

Call sendNotification() on each object in the array to demonstrate the functionality of sending notifications.

Expected Output:-

The program should display the messages for both email and SMS notifications when executed. For example:

Sending Email: Welcome to our service!

Sending SMS: Your verification code is 123456.

Constraints:-

Ensure proper encapsulation by keeping the message field private in both classes.

Follow good coding practices, including meaningful variable names and comments.

Make sure to handle potential null values for the message in the sendNotification() method.

**5) Media Player System with Abstract Class and Interfaces**

Objective: Implement an abstract class and interfaces to demonstrate a media player system capable of playing and stopping different types of media.

Requirements:

Define an Abstract Class:

Create an abstract class named Media with an abstract method play().

Include a method stop() in the Media class that provides a default implementation for stopping media playback.

Define Interfaces:

Create an interface named Playable with a method play().

Create an interface named Stoppable with a method stop().

Implementing Classes:

Create a class named Song that:

Extends the Media class.

Implements the Playable and Stoppable interfaces.

Provides its own implementation of the play() method to display a message like: "Playing song: [song name]".

Implements the stop() method to display a message like: "Stopping song: [song name]".

Create a class named Video that:

Extends the Media class.

Implements the Playable and Stoppable interfaces.

Provides its own implementation of the play() method to display a message like: "Playing video: [video title]".

Implements the stop() method to display a message like: "Stopping video: [video title]".

Demonstration:

In the main method of a separate class, create an array of Media objects.

Create instances of Song and Video, and add them to the array.

Call play() and stop() methods for each object in the array to demonstrate the functionality of playing and stopping different media types.

Expected Output:

The program should display messages indicating the playback and stopping of both songs and videos when executed. For example:

Playing song: Imagine song playing(give appropriate message)

Stopping song: Imagine song stoped(give appropriate message)

Playing video: Inception(give appropriate message)

Stopping video: Inception(give appropriate message)

Constraints:

Ensure proper encapsulation by managing the fields required for song and video titles.

Follow good coding practices, including meaningful variable names and comments.

Ensure that the stop() method in the Media class can be overridden by subclasses if needed.

**6) Transportation System with Abstract Class and Interface**

Objective: Implement an interface and an abstract class to create a transportation system that allows different types of vehicles to move and stop.

Requirements:

Define an Interface:

Create an interface named Transport with the following methods:

void move(); - for moving the transport.

void stop(); - for stopping the transport.

Define an Abstract Class:

Create an abstract class named Vehicle that:

Implements the Transport interface.

Leaves the move() method abstract, requiring subclasses to provide their own implementation.

Provides a concrete implementation of the stop() method that displays a message indicating the vehicle has stopped.

Implementing Classes:

Create a class named Car that:

Extends the Vehicle class.

Implements the move() method to display a message like: "The car is moving."

Create a class named Bus that:

Extends the Vehicle class.

Implements the move() method to display a message like: "The bus is moving."

Demonstration:

In the main method of a separate class, create an array of Transport objects.

Create instances of Car and Bus, and add them to the array.

Call the move() and stop() methods for each object in the array to demonstrate the functionality of moving and stopping different transport types.

Expected Output:

The program should display messages indicating the movement and stopping of both the car and the bus when executed. For example:

The car is moving.

The bus is moving.

The car has stopped.

The bus has stopped.

Constraints:

Ensure proper encapsulation by managing the fields required for vehicle attributes.

Follow good coding practices, including meaningful variable names and comments.

Make sure the stop() method in the Vehicle class can be overridden by subclasses if needed.

**7) Payment Gateway System**

Objective: Implement a system for processing different types of payments.

Requirements:

Define an Interface:

Create an interface named PaymentMethod with methods processPayment(double amount) and refundPayment(double amount).

Define an Abstract Class:

Create an abstract class named BasePayment that implements PaymentMethod.

Provide a concrete implementation of refundPayment() to display a message indicating the refund process.

Leave processPayment(double amount) abstract.

Implementing Classes:

Create a class CreditCardPayment that extends BasePayment and implements processPayment() to process credit card payments.

Create a class PayPalPayment that extends BasePayment and implements processPayment() to process PayPal payments.

Demonstration:

Create an array of PaymentMethod objects.

Initialize it with instances of CreditCardPayment and PayPalPayment.

Call processPayment() and refundPayment() for each payment method to demonstrate functionality.

Expected Output:

Processing payment of $100.00 via Credit Card.

Refunding payment of $50.00 via Credit Card.

Processing payment of $75.00 via PayPal.

Refunding payment of $30.00 via PayPal.

**8) Recipe Organizer System**

Objective: Create a system for managing different types of recipes.

Requirements:

Define an Interface:

Create an interface named Recipe with methods prepare() and getIngredients().

Define an Abstract Class:

Create an abstract class named BaseRecipe that implements Recipe.

Provide a concrete implementation of getIngredients() to display a generic list of ingredients.

Leave prepare() abstract.

Implementing Classes:

Create a class VegetableRecipe that extends BaseRecipe and implements prepare() to provide instructions for preparing a vegetable dish.

Create a class DessertRecipe that extends BaseRecipe and implements prepare() to provide instructions for preparing a dessert.

Demonstration:

Create an array of Recipe objects.

Initialize it with instances of VegetableRecipe and DessertRecipe.

Call prepare() and getIngredients() for each recipe to demonstrate functionality.

Expected Output:

Preparing a delicious vegetable stir-fry.

Ingredients: vegetables, oil, spices.

Preparing a rich chocolate cake.

Ingredients: flour, sugar, cocoa powder, eggs.

**9) Multi-Channel Notification System**

Objective: Implement a notification system that supports different channels of communication.

Requirements:

Define an Interface:

Create an interface named Notifiable with methods sendNotification(String message) and setRecipient(String recipient).

Define an Abstract Class:

Create an abstract class named Notification that implements Notifiable.

Leave both methods abstract.

Implementing Classes:

Create a class EmailNotification that extends Notification and implements both methods to send email notifications.

Create a class SMSNotification that extends Notification and implements both methods to send SMS notifications.

Demonstration:

Create an array of Notifiable objects.

Initialize it with instances of EmailNotification and SMSNotification.

Set the recipient and send a notification for each object to demonstrate functionality.

Expected Output:

Sending email to user@example.com: Your order has been shipped!

Sending SMS to 123-456-7890: Your order has been shipped!

**10) Employee Management System**

Objective: Create a system to manage different types of employees in a company.

Requirements:

Define a Base Class:

Create a class named Employee with attributes such as name, id, and salary.

Include a method displayInfo() that prints employee details.

Implement Subclasses:

Create a class Manager that extends Employee and adds an attribute for department.

Override displayInfo() to include department details.

Create a class Intern that extends Employee and adds an attribute for duration (in months).

Override displayInfo() to include duration details.

Demonstration:

Create an array of Employee objects.

Initialize it with instances of Manager and Intern.

Call displayInfo() for each employee to demonstrate functionality.

Expected Output:

Name: Alice, ID: 001, Salary: $70000, Department: Sales

Name: Bob, ID: 002, Salary: $30000, Duration: 6 months

**11) Course Management System**

Objective: Create a system to manage different types of courses in an online learning platform.

Requirements:

Define a Base Class:

Create a class named Course with attributes such as courseName, duration, and instructor.

Include a method getDetails() that prints course details.

Implement Subclasses:

Create a class VideoCourse that extends Course and adds an attribute for videoLength.

Override getDetails() to include video length.

Create a class TextCourse that extends Course and adds an attribute for numberOfPages.

Override getDetails() to include the number of pages.

Demonstration:

Create an array of Course objects.

Initialize it with instances of VideoCourse and TextCourse.

Call getDetails() for each course to demonstrate functionality.

Expected Output:

Course Name: Java Programming, Duration: 4 weeks, Instructor: John Doe, Video Length: 10 hours

Course Name: Data Structures, Duration: 3 weeks, Instructor: Jane Smith, Number of Pages: 120

**12) Fitness Activity Management**

Objective: Create a system to manage different types of fitness activities.

Requirements:

Define a Base Class:

Create a class named Activity with attributes such as activityName, duration, and caloriesBurned.

Include a method getSummary() that prints activity summary.

Implement Subclasses:

Create a class Running that extends Activity and adds an attribute for distance.

Override getSummary() to include distance details.

Create a class Cycling that extends Activity and adds an attribute for speed.

Override getSummary() to include speed details.

Demonstration:

Create an array of Activity objects.

Initialize it with instances of Running and Cycling.

Call getSummary() for each activity to demonstrate functionality.

Expected Output:

Activity: Running, Duration: 30 min, Calories Burned: 300, Distance: 5 km

Activity: Cycling, Duration: 45 min, Calories Burned: 400, Speed: 15 km/h

**24/10/2024**

**Exception handlling, String, file handling Assignment**

Q1. What is exception and why use it?

Q2. Explain exception handling hierarchy?

Q3. Explain benefits of exception handling?

Q4. Explain types of exception in java?

Q5. What is diff between exception and error?

Q6. Explain keyword for exception handling with examples?

Q7. What is diff between catch and finally block?

Q8. What is throws keyword in exception handling and why use it?

Q9. What is throw keyword in exception handling and why use it?

Q10. What is diff between throws and throw keyword?

Q11. Explain any 10 exceptions with example?

Q12. Can we write try without catch block?

Q12. What is try with resource bundle?

Q13. Can we write more than one exception in catch block using pipe operator?

Q14. Can we write try within try block?

Q15. Can we write multiple catch for single try?

Q16. Is it true throw and throws throw exception object at function calling point?

Q17. Is it true throws design for handle checked exception

Q1. What is wrapper classes in java and why use it?

Q2. Explain type casting and its type?

Q3. What is implicit type casting and explicit type casting?

Q4. Explain hierarchy of wrapper classes in java?

Q5. Explain Number and class and its child classes with its method?

Q6. What is autoboxing and auto unboxing in wrapper classes?

Q7. Explain use of xxxValue() method of Number class with example?

Q8. Explain valueOf() method of Wrapper classes?

Q9. What is parseXXX() method of Wrapper classes?

Q10. what is String in java explain in depth?

**String Question**

Q11. What is meaning of immutable in java?

Q12. How many ways to create string in java?

Q13. What is diff between string creation using “ “ and using a new keyword?

Q14. What is string constant pool ?

Q15. Explain any 4 constructor of string class constructor?

Q16. Explain charAt(),length(),indexOf(),substring(),split(),trim()

File Handlling

Q1. what is streams in java and why use it?

Q2. how to many ways to work with file in java?

Q3. Explain File class from java.io package?

Q4. Explain how to get all drive names, total space of drive, free space of drive etc?

Q5. Explain Hierarchy of OutputStream class?

Q6. Explain Hierarchy of Writer class?

Q7. Explain Methods of Writer class?

Q8. Explain Methods of OutputStream class?

Q9. Explain FileWriter class with its constructor and write code for storing data in .txt file?

Q10. Explain BufferedWriter class with example?

Q11. what is diff between FileWriter and BufferedWriter class?

Q12. Explain FileReader and BufferedReader with example?

Q13. Explain ObjectOutputStream class?

Q14. Explain ObjectInputStream class?

Q15. what is Serialization and Deserialization in JAVA?

Q16. Write code for serialization and deserialization in java?

Q17. Write code for reading .csv files in java?

Programs

1. Write a JAVA program to count total number of words in a string.

2. Write a JAVA program to find reverse of a string.

3. Write a JAVA program to check whether a string is palindrome or not.

4. Write a JAVA program to reverse order of words in a given string.

5. Write a JAVA program to find first occurrence of a character in a given string.

6. Write a JAVA program to find last occurrence of a character in a given string.

7. Write a JAVA program to search all occurrences of a character in given string.

8. Write a JAVA program to count occurrences of a character in given string.

9. Write a JAVA program to find highest frequency character in a string.

10. Write a JAVA program to find lowest frequency character in a string.

11. Write a JAVA program to count frequency of each character in a string.

12. Write a JAVA program to remove first occurrence of a character from string.

13. Write a JAVA program to remove last occurrence of a character from string.

14. Write a JAVA program to remove all occurrences of a character from string.

15. Write a JAVA program to remove first occurrence of a word from string.

File Handling

1. Write a program in java to create and store information in a text file.

Test Data :

Input a sentence for the file : This is the content of the file test.txt.

Expected Output :

The file test.txt created successfully...!!

2. Write a program in java to read an existing file.

Test Data :

Input the file name to be opened : test.txt

Expected Output :

The content of the file test.txt is :

This is the content of the file test.txt.

3. Write a program in java to write multiple lines to a text file.

Test Data :

Input the number of lines to be written : 4

:: The lines are ::

test line 1

test line 2

test line 3

test line 4

Expected Output :

The content of the file test.txt is :

test line 1

test line 2

test line 3

test line 4

4. Write a program in java to read the file and store the lines in an array.

Test Data :

Input the file name to be opened : test.txt

Expected Output :

The content of the file test.txt are :

test line 1

test line 2

test line 3

test line 4

5.Write a program in java to find the number of lines in a text file.

Test Data :

Input the file name to be opened : test.txt

Expected Output :

The lines in the file test.txt are : 4

**25/10/2024**

**Collection Framework, Thread, Jdk1.8**

Q1. What is Collection and why use Collection Framework explain with 5 reason?

Q2. What are benefit of Collection over array?

Q3. Explain Collection Hierarchy?

Q4. Explain Iterable interface in java with its method?

Q5. What is Collection interface and explain all methods of Collection interface?

Q6. Explain add() method, Boolean contain() method,indexOf() method,size(),Boolean

isEmpty(),remove(),iterator() method etc

Q7. Explain types of Collection like as List,Set and Queue interface?

Q8. What is diff between List an Set interface?

Q9. What is diff between List and Queue interface?

Q10. Explain all implementer classes of Collection framework?

Q11. Explain Vector class with its constructor?

Q12. Explain any 5 important points of Vector class?

Q12. Is it true Vector is synchronized collection and explain its reason?

Q13. Is it true vector is legacy collection?

Q14. What is legacy collection in java?

Q15. What is default capacity of Vector?

Q16. Is it true Vector occupy double memory than its current capacity?

Q17. Explain logic of Vector double capacity increment?

Q18. What is diff between Vector and ArrayList?

Q19. What is default capacity of Vector and ArrayList collection?

Q20. How ArrayList increase its capacity when capacity cross?

Q21. What is diff between LinkedList and ArrayList?

Q22. What is time complexity of LinkedList?

Q23. What is time complexity of ArrayList?

Q24. Who is better for element deletion or updation between LinkedList and ArrayList?

Q25. Who is better for data fetching or data retrieval between LinkedList and ArrayList?

Q26. What is threads hold value of Vector?

Q27. Explain stack in depth?

Interview Question on Map interface in java?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Q1. What is Map?

Q2. Explain Hierarchy of Map interface?

Q3. Explain any 8 methods of Map interface?

Q4. Explain HashMap in depth?

Q5. Explain LinkedHashMap in depth?

Q6. What is diff between HashMap and LinkedHashMap interface?

Q7. What is NavigableMap interface?

Q8. What default load factory of HashMap in java?

Q9. Explain internal working of HashMap?

Q10. Explain Map.Entry in java

**program**

1. How To Sort HashMap In Java By Keys And Values

2.Write a java program to sort an ArrayList?

3.Write a program Find the maximum and minimum program in ArrayList?

4.write a program Take a ArrayList add 5 Employee Object and sort data by salary?

5. How To Iterate Or Loop Over HashMap (Map) In Java With Example?

7. How To Convert Array To TreeSet In Java?

8.find the character of occurrence using LinkedHashMap?

9. Write a program to find the numbers of occurrences in string? - by providing many strings using Map ?

10.Remove The Duplicate value from List?

Q11. WAP to create Vector and store 5 values in it and find max and min value from collection without using any inbuilt method?

Q12. WAP to create ArrayList and store 10 values in it and find occurrence of every value in ArrayList?

Q13. WAP to create LinkedList and store 10 values in it and search particular value from LinkedList? Q14.WAP to create ArrayList and store 5 employee objects in it and display it?

Q15. WAP to create ArrayList and store 5 player detail in it with id,name and run and search player record using its id.

Q16. WAP to create HashSet and store 10 value in it and display it

Q17. WAP to create LinkedHashSet and store 5 employee data in it and display it ?

Q18. WAP to create TreeSet and store 10 values in it and display it?

Q 19and store value in it and find the occurrence of every element using LinkedHashMap

Q20. WAP to create input string and split its word and find the repetitive word in string using LinkedHashMap

Q 21. Write a java code in which you have to find the intersection from two hashset

Q22.Write a java code in which you have to create unmodified list collection

Q23. Write a Java Code in which you have to create a Syncronized Set Collection .

Q23 . Write a Java code in which you have to find the Maximum Value and Key From HashMap

Q 24.Write a Java Code in Which you have to Find The Last Key ANd Value From TreeMap

Q 25.Write A JAva Code in WHich you Have To find Frist Key And Value From TreeMap

Q .26 Write a Java code in which you have to move all zero value at end of array

Ex.arr[]={11,0,22,44,66,0,99,0}

Output Required : arr[]{11,22,44,66,99,0,0,0};

Q 27. Write a Java program to insert elements into the linked list at the first and last positions.

Q28 .Write a Java program to remove the first and last elements from a linked list.

Q29. Write a Java program to copy all mappings from the specified map to another map.

Q30 Write a Java program to get a collection view of the values contained in this map.

**Thread**

Q16. What is thread priority in java and explain all types of thread priority?

Q17. What is daemon thread in java?

Q18. Can we create user thread as daemon thread?

Q19. Why we need to implement thread using a Runnable interface? Q20 Write code for creating thread using Runnable interface

**26/10/2024**

**HTML, CSS**

1. Create an HTML file (e.g. first\_page.html) that specifies a page that

contains a heading and two Paragraphs of text. Use the HTML tags <h1>,

</h1>, <p> </p>,<div> and </div> in this exercise. As the texts in the

Heading and paragraphs, you can use any text you like.

-Add an unordered list to this web page. An unordered list should look l

like the following when it is shown by a browser:

- An unordered list can be specified with the tags <ul> and </ul>.

- An unordered list typically contains several list items that can be

Specified With tags <li> and </li>.

-After you have created your unordered list, check out what happens

when you Convert it to an ordered list by replacing the tags <ul> and

</ul> with <ol> and </ol>, respectively.

-Add an image to your web page. In this exercise, you must use the

<img> tag. As an image, you can use any .jpg or .png file you find on the

Internet.

2. Create another .html file that contains a heading and a couple of

paragraphs. You could name this new file another\_page.html, and you

should place it into the same folder where your first .html is. After you

have created the new .html page, add a link to the first page so that the

browser will load another\_page.html when you click the text Go to the

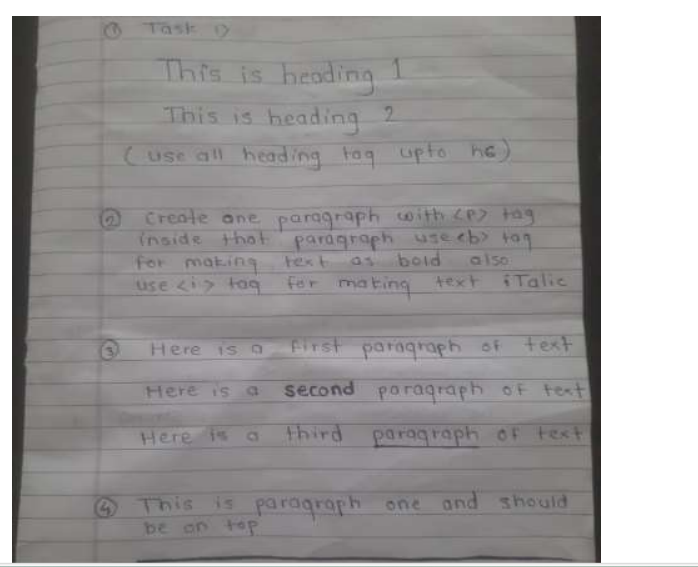
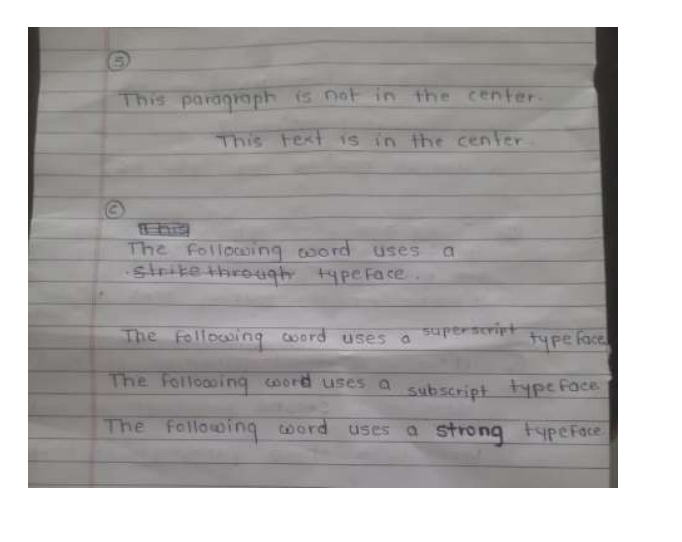
other page. on the first page. You need to use the <a> and </a> tags in

this exercise. Inside the tag <a> you need to use a href attribute that

specifies which page will be loaded when the link is clicked also use the

tooltip for <a> tag that says "This leads you to another page." when the

mouse cursor is over the link.

1. Completed the assignment as mentioned below image
2. Use the image map concept and perform the following task -When we click on Mouse, Diary, Keyboard, or Pen it will redirect to another webpage that tells the information about that device.



2. Use the Audio and Video tag concept and perform the following task

- Use the below audio link and demonstrate an Audio Tag

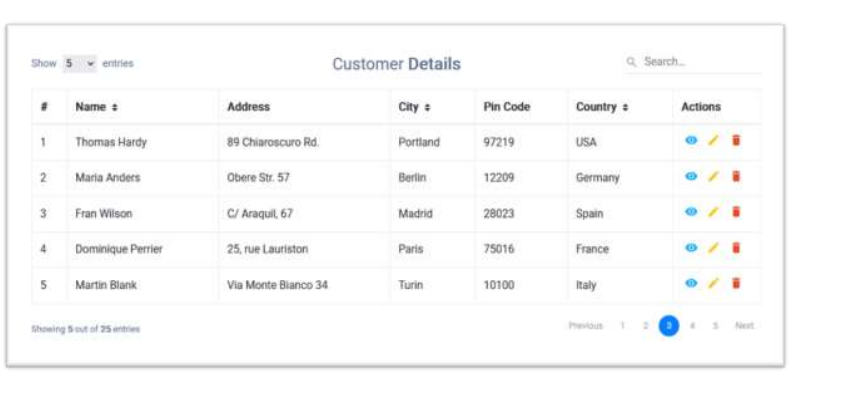
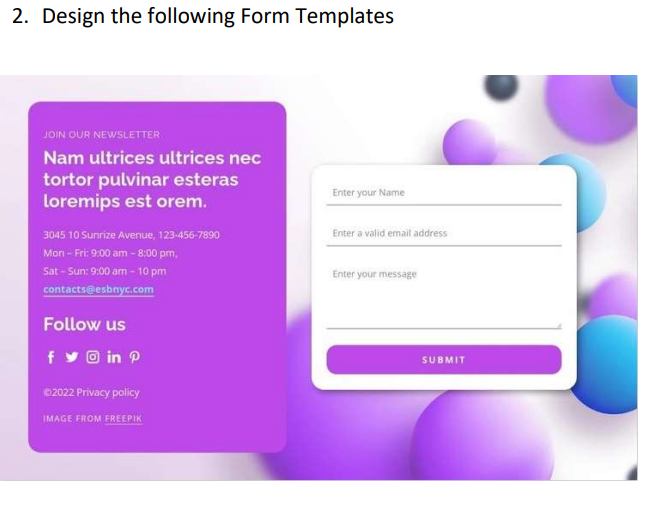
Audio Link:- https://samplelib.com/lib/preview/mp3/sample-3s.mp3

- Use the below Video link and demonstrate a Video Tag

Video Link:- https://media.w3.org/2010/05/sintel/trailer.mp4

3. Design the following Table template

Html with CSS 16



**27/10/2024**

**JavaScript**

Q. what is diff between JavaScript and Java?

Q. what is java script and why use it?

Q. how many ways to use JavaScript?

Q. what is internal and external java script?

Q. what is use of typeof() operator

Q. what is event and explain events in javascript?

Q. what is function and how to write function in javascript?

Q. what is dynamic type language and is java script is dynamic type language?

Q. what is diff between static and dynamic type language?

Q. what is DOM and explain DOM Tree?

Q. explain any 5 methods of DOM?

Looping Program

1. Javascript Program to check if a number is Positive, Negative, or Zero

2. JavaScript Program to Check Prime Number

3. JavaScript Program to Find the Factorial of a Number

4. JavaScript Program to Display the Multiplication Table

5. JavaScript Program to Check Armstrong Number

Array Program

1. JavaScript Program to Insert Item in an Array

2. JavaScript Program to Add Element to Start of an Array

3. JavaScript Program to Remove Duplicates From Array

4.WAP to enter the five elements in array and find out the maximum and minimum elements

5. Write a program in C to merge two arrays of same size sorted in decending order.

String

1. Write a JAVA program to convert lowercase string to uppercase.

2. Write a JAVA program to find total number of alphabets, digits or special character in a string

3.Write a JAVA program to count total number of words in a string

4. Write a JAVA program to check whether a string is palindrome or not

5.Write a JAVA program to find first occurrence of a character in a given string

Dom

1. Write JavaScript program to design following form and Design calculator shown given below

Your task is input two values in first and second textbox and when we click on ADD button

then calculate addition and show in result text box, when we click Sub button

then show result in result box and when we click on MUL then show result in third text box.

2.Write JavaScript program to design we want to create button using HTML and when we click on

button then we want To call java script function on button click and show message welcome in

event handling using alert box.

3.Write JavaScript program we want to design web page with one heading and one button and when we click on button we want to access heading tag data in java script and display in

alert box

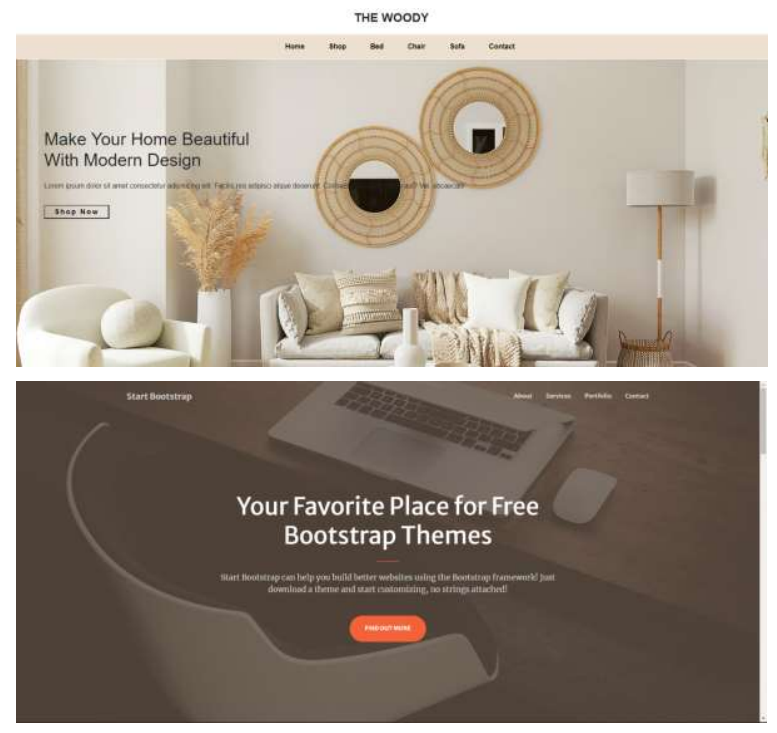
4.Write JavaScript program we want to add text in h1 tag when we click on button

means at initial level we want to show good morning india in heading but after click we want to show good afternoon india in heading means we want to replace data good afternoon india on good morning india.

1.Design the following Form Template and demonstrate background image

properties.

**BootStrap**





29/10/2024

**29/10/2024**

**SQL: DDL,DML,TCL,DCL,Constraints,Operator,Aggregation Function**

Q1. what is database and why use it?

Q2. explain feature of database?

Q3. what is diff between mysql database and SQL?

Q4. explain types of database?

Q5. what is DDL and DML commands and explain in details?

Q6. what is constraints and why use constraints?

Q7. explain types of constraints

Q8. what primary key and foreing key and its differences.

Q9. what is diff between unique and primary key

Q10. what is diff betweeen unique key and foreign key

Q11. what is candidate key

Q12 what is composite primary key

q13 what is joins and explain types of join with example

Q14. what is normalization and explain its type

Q15 what happen if we design database without normalization

Assignment

**Task 1**

Employee Table:

| EmployeeID | FirstName | LastName | DepartmentID | Salary | HireDate |

|------------|-----------|----------|--------------|--------|-----------|

| 1 | John | Smith | 1 | 55000 | 2021-05-15|

| 2 | Emily | Johnson | 2 | 60000 | 2020-12-10|

| 3 | Michael | Davis | 1 | 48000 | 2022-02-28|

| 4 | Sarah | Wilson | 2 | 51000 | 2021-08-20|

| 5 | David | Clark | 1 | 58000 | 2022-01-03|

| 6 | Jessica | Lee | 3 | 52000 | 2021-04-12|

| 7 | Brian | Hall | 3 | 49000 | 2020-07-22|

| 8 | Amy | Adams | 2 | 56000 | 2022-03-10|

| 9 | Kevin | Turner | 1 | 53000 | 2020-11-05|

| 10 | Laura | White | 3 | 47000 | 2022-04-18|

Company Table:

| CompanyID | CompanyName | FoundedYear | CEO | Location |

|-----------|-------------|-------------|---------------|-----------------|

| 1 | XYZ Corp | 1995 | John Johnson | New York |

| 2 | ABC Inc | 2000 | Sarah Wilson | Los Angeles |

| 3 | QRS Ltd | 1988 | Michael Davis | Chicago |

| 4 | LMN Co | 1992 | Emily Johnson | San Francisco |

| 5 | PQR Co | 2005 | David Clark | Boston |

1. Create a new table named `Projects` with columns ProjectID (Primary Key), ProjectName, DepartmentID (Foreign Key referencing Department table), ProjectStartDate, and ProjectEndDate.

2. Add a foreign key constraint on `DepartmentID` in the `Projects` table, referencing `DepartmentID` in the `Department` table.

3. Add a unique constraint on the combination of `FirstName` and `LastName` in the `Employee` table.

4. Add a default constraint to the `Salary` column in the `Employee` table, setting it to 45000.

5. Add a check constraint on `Salary` in the `Employee` table, ensuring it is between 30000 and 70000.

6. Create a non-clustered index on the `CEO` column in the `Company` table.

7.Create a sequence named `EmployeeIDSeq` starting from 1001 and incrementing by 1.

8.Create a unique clustered index on `ProjectID` in the `Projects` table.

9. Rename the `CEO` column in the `Company` table to `ChiefExecutiveOfficer`.

10. Truncate all records from the `Projects` table.

11.Insert a new project into the `Projects` table with appropriate values for all columns.

12. Update the `Salary` of all employees in the 'Sales' department to 48000.

13.Delete all projects that have an end date before 2022-01-01.

14. Increase the `Salary` of employees hired before 2021 by 10%.

15. Delete employees whose last names start with 'S' and have a salary less than 55000.

16. Find the projects associated with employees whose salary is greater than the average salary of all employees.

**30/10/2024**

**Joins,SubQuries,Views,Index,Procedure,Trigger**

Q1. what is depenendency and explain types of dependency

Q2. explain partial dependency and transitive dependency

Q3 explain in,between,exists operations

Q4. what is sub query and explain types of sub query

Q5. what is diff between join and sub query

Q6. what is used for group by ,having, and order by and like and where

Q7 what is indexing and explian types of indexing

Q8 what is procedure and explain in dept

Q9. what is function and what is diff between function and procedure

Q10. what is tringger and explain types of trigger?

Q11. what is diff between procedure and tringger?

Q12. what is diff between delete and truncate

Q13. what is diff between on delete cascade and on delete set null and on delete update

Q14. what is cascading and why use it

Q15 explain string handling function in mysql

Q16. explain date and time api in mysql

Q17. what is cursor and why use cursor in mysql.

Q18. what is accid property and explian in details.



**02/11/2024**

**JDBC,Servlet**

1. What is JDBC?
2. What is JDBC Driver?
3. What are the steps to connect to the database in java?4
4. What are the JDBC API components?
5. What are the JDBC statements?
6. What are the differences between execute, executeQuery, and executeUpdate?
7. What are the different types of ResultSet?
8. How can we execute stored procedures using CallableStatement?
9. What is the role of the JDBC DriverManager class?
10. What are CLOB and BLOB data types in JDBC?
11. What are the differences between stored procedure and functions?
12. What is the major difference between java.util.Date and java.sql.Date data type?

Q. what is servlet and why use it?

Q. what is dynamic web page and how to create dynamic web page using a servlet?

Q. what is port number and why use it?

Q.what is server and explains application server and web server?

Q. what is diff between generic servlet and HttpServlet?

Q. what is diff between doGet and doPost method?

Q. explain servlet life cycle?

Q. how to submit form in servlet?

Q. what is RequestDispatcher interface and explain in depth?

Q. what diff between sendRedirect() and forward() method?

Q. what is a session and why use it ?

Q. explain ways of session handling ?

Q. what is cookies and why use it?

Q. what is Servletcontext and ServletConfig?

Q. diff between ServletContext and ServletConfig?

Q Why use of getInitParameter?

**JDBC Assignments**

1. **Practical**: Write a JDBC program to establish a connection to a MySQL database and retrieve the names of all tables present in the database.
2. **Practical**: Create a JDBC application that allows a user to insert a new record into a Students table. Prompt for student details like name, email, and contact number.
3. **Practical**: Develop a JDBC program that updates the email address of a student based on their student ID. Implement error handling for cases where the student ID does not exist.
4. **Practical**: Write a JDBC application that deletes a record from the Students table based on user input for the student ID.
5. **Practical**: Create a JDBC program that retrieves all records from the Courses table and displays them in a formatted manner in the console.

**JSP Assignments**

1. **Practical**: Develop a JSP page that accepts user input (e.g., name, email, and contact number) via a form. Use JSTL to display the submitted information on a new JSP page.
2. **Conceptual**: Create a JSP application that demonstrates session management by allowing users to log in and store their name in a session variable. Display a welcome message on a separate JSP page after login.
3. **Practical**: Build a JSP page that connects to a database using JDBC to display all records from the Courses table in an HTML table format.

**Servlet Assignments**

1. **Practical**: Write a simple servlet that responds to HTTP GET requests by displaying a welcome message in the browser.
2. **Practical**: Create an HTML form that accepts user information (name, email). Write a servlet to handle the form submission and display the entered data on a new webpage.

Let me know if you need any more details!

**03/11/2024**

**Jsp**

Q. What is JSP and why use it?

Q. What is diff between Servlet and JSP?

Q. Explain life cycle of JSP?

Q. Explain all tags of JSP?

Q. Explain all objects of JSP?

Q. what is action tag and explain all action tags of JSP?

Q. what is MVC and why use it?

Desing CRUD Application using Servlet and jsp

Student Management Application using Servlet and jsp And Appropriate fields(Id,Name,Email,Contact ,Addrsss,Date)

We will develop CRUD Feature

1. Add New Student

2. View All Student

3. Update Student

4. Delete Student.

5.Search Student by using Name (Using Ajax)

**04/11/2024**

**Spring Core**

***Example1: WAP to create spring core which display Hello World Application***

***Example2: WAP spring application create class name as Employee with id name and sal and perform***

Setter injection using XML Configuration as well as perform setter injection using annotation?

***Example3: WAP to create spring core application with class name as Employee with parameterized***

constructor and Employee class constructor contain three parameter id, name and sal and perform

constructor injection with Employee class and also give the index of every parameter

**Example4: WAP to Create spring core application with two classes name as Player and Team**

**Player is POJO class which contain three field id, name and run and we have Team class which**

**contain two methods**

void setPlayer (Player player): this method accepts Player class object using setter method means

setter injection

void showPlayer (): this method can display the player information

Note: you have to perform object dependency using setter injection technique in spring core using

XML as well as using @Autowired annotations

**Example5: WAP to Create spring core application with two classes name as Player and Team**

**Player is POJO class which contain three field id, name and run and we have Team class which**

**contain two methods**

Team (Player player): this method accepts Player class object using constructor method means

constructor injection

void showPlayer (): this method can display the player information

Note: you have to perform object dependency using setter injection technique in spring core.

**Example6: Create Spring core Application for manage the collection dependency using list collection**

**means you have class name as Shop with two methods**

void setProducts (List<String> list): this method accepts all product using list collection

void showList (): this method can display the all products from list collection

Note: Perform collection dependency using XML.

**Example7: Create Spring core Application for manage the collection dependency using set collection**

**means you have class name as Shop with two methods**

void setProducts (Set<String> set): this method accepts all product using set collection

void showList (): this method can display the all products from list collection

Note: Perform collection dependency using XML

**Example8: Create Spring core Application for manage the collection dependency using map**

collection means you have class name as Shop with two methods

void setProducts (Map<Integer,String> map): this method accepts all product using map

dependency means you have to pass data of product in the form of key and value pair to setter

function.

void showList (): this method can display the all products from list collection

Note: Perform collection dependency using XML

**Example9: WAP to Create spring core application with two classes name as Player and Team**

Player is POJO class which contain three field id, name and run and we have Team class which

contain two methods

void setPlayer (Player player): this method accepts Player class object using setter method means

setter injection

void showPlayer (): this method can display the player information

Note: in this application you have to use auto wiring technique using XML as well as using

@Autowired annotations.

**Example10: WAP to create class name as Test and set its scope as singleton using XML as well as**

**using annotations**

**Example11: WAP to create class name as Test and set its scope as prototype using XML as well as**

**using annotations.**

**05/11/2024**

**Spring JDBC**

1. What is spring JDBC
2. Why use spring JDBC
3. What is JDBC Template and explain in dept
4. Write a program for select record from a database using Spring JDBC
5. Write a program to insert record in database in using spring JDBC
6. What is PreparedStatementSetter and RowMapper
7. What is diff between normal JDBC and Spring JDBC

Spring mvc

1.What is MVC

2.What is spring MVC

3.Explain Spring MVC Architecture

4.What is Controller and how to create it

5.What is RequestMapping and how to write it

6.What is view and how to write view I n Spring MVCWrite a code for create controller and call view page

7.How to send data from controller to view

8.What is model and why use it

9.Write a code for save record in database by using spring MVC

10.Why use spring MVC we have servlet and jsp

11.hat is spring boot why use it

12 What is meaning of @SpringBootApplication

13 What is RestController

14 Explain methods of rest api

15 What is postman and why use

16 What is maven and why use it

17 What is @RequestBody ,@ResponseEntity annotation What is json and how use it

18 What is web services and why use it

19 What is restful web service

20 What is used json and xml

21 Write a code for create json object and store data in it and display it

22 What is thymeleaf and why use it What is JPA and why use it

21 What is diff between hibernate and JPA

23 What is CrudRepository and JPA Repository and explain its difference

24 What is finder method and how to write it

25 How to write custom query in JPA

26 What is used of @Query ,@Param annotations

27 What is used for @Service. @Repository annotations

Desing CRUD Application spring mvc

Student Management Application Spring mvc Appropriate fields(Id,Name,Email,Contact ,Addrsss,Date)

We will develop CRUD Feature

1. Add New Student

2. View All Student

3. Update Student

4. Delete Student.

5.Search Student by using Name (Using Ajax)

**07/11/2024**

**Hibernet**



**Assignment Question**

**1. Simple CRUD with Hibernate**

Create a Hibernate-based application to perform CRUD operations on a Product entity. The Product entity should have fields: id, name, description, price. Write methods to:

* Create a new product
* Read a product by its id
* Update the product details
* Delete a product

**2. Hibernate Mapping – One-to-One**

Design a Hibernate application that models a one-to-one relationship between User and Address. A User should have fields: id, name, email, and a Address with fields: id, street, city, state, zipcode. Implement CRUD operations for User, ensuring the Address is handled automatically by Hibernate.

**3. Hibernate Mapping – One-to-Many**

Model a one-to-many relationship using Hibernate between Author and Book. An Author can have multiple Books. The Author entity should have fields: id, name, email, while Book should have fields: id, title, isbn, price. Implement a method to create an author with a list of books and retrieve all books by a specific author.

**4. Hibernate Mapping – Many-to-Many**

Create a many-to-many relationship using Hibernate between Student and Course. A Student can enroll in multiple Courses and each Course can have multiple Students. The Student entity should have id, name, email, while the Course entity should have id, title, description. Implement CRUD operations and a method to list all students enrolled in a particular course.

**5. Hibernate Query Language (HQL)**

Use HQL to create an application that queries data from the database. Using the Employee entity (fields: id, name, department, salary), write the following queries:

* List all employees
* Find an employee by their id
* Find all employees in a specific department
* Find employees with a salary greater than a specified amount
* Update the salary of employees in a certain department

**Spring Boot + React 8/11/2024**

1 What is Spring Boot, and how is it different from Spring Framework?

2 What is the purpose of the @SpringBootApplication annotation?

3 What is auto-configuration in Spring Boot?

4 How do you run a Spring Boot application?

5 What is Spring Boot Starter, and why is it useful?

6 What are the main features of Spring Boot?

7 What is the difference between @RestController and @Controller?

8 How do you configure properties in Spring Boot, and what is the difference between application.properties and application.yml?

9 How do you define and access custom properties in Spring Boot?

10 What are Spring profiles, and how do you use them?

11 How does Spring Boot handle dependency management?

12 What is @ConfigurationProperties, and how is it different from @Value?

13 How does Spring Boot handle logging, and how do you configure it?

14 What is the role of an embedded server in Spring Boot?

15 How do you handle exceptions in Spring Boot using @ControllerAdvice and @ExceptionHandler?

**9/11/2024**

